EEL4713 Assignment #4
Spring 2014
Final Demo: Thursday 11/13/14 in Lab
Report Due: Saturday 11/15/14 @ 11:55pm Via Sakai
Developed by Scott Arnold and Ann Gordon-Ross Fall 2012

Overview;
In this assignment, you will be designing a multi-cycle implementation of a MIPS processor without hazard detection. Your processor will need to be able to successfully execute Lab3.mif in the same way that your single-cycle processor did—all of the instructions listed under the Core Instruction Set on the green reference sheet from your book, except for ll (load linked) and sc (store conditional). You must be using the mif file given in the previous lab to demonstrate the functionality of your processor for the in lab demo. Note: If you are still having errors in your single-cycle processor, you must fix those errors before this assignment.

In this assignment, you will perform the following tasks:

DEMOS: final demo of working processor
2.1: Design and simulate a MIPS multi-cycle processor
2.2: Annotate the simulations for the Lab3.mif demo program
2.3: Create the hazard detection table
3: Review deliverables and write the report

Section 1: Setup
There is no setup in this assignment.

Section 2: Laboratory
2.1: Multi-Cycle Processor
In this assignment, you will be modifying your single-cycle processor from Assignment 4 in order to implement a pipelined processor. Your processor will have to successfully execute the 29 instructions listed on the front of the green reference sheet, but will not need to detect hazards, and forward the needed data in order to correct the detected hazards at this point. Please read sections 4.6-4.8 of your textbook for insight on how this can be done.

Begin your design by partitioning your single-cycle processor into 5 stages: instruction fetch (IF), instruction decode (ID), execution (EX), memory access (MEM), and write back (WB). Consult figures 4.28, 4.29, and 4.33 in your textbook for hints on how you should partition your design. Next, create four pipeline registers to place in between your pipeline stages that will transfer the needed data from stage to stage. It is possible to eliminate the "mclk" signal by using the registered input of the altsyncram component as a register between the pipeline stages. Eliminating the "mclk" signal should greatly increase your maximum clock rate and is highly recommended.

At this point your processor should be able to execute hazard-free code without any problems.

2.2: In-Lab DEMOS
Final Demo: Demonstrate the functionality of your processor using the Lab3.mif file you generated from your assembler in Assignment 2 or the one given in the previous class.

Include a section in your report where you annotate the functional simulation output of your processor running the program.

Timing analysis of your processor should be done with the **Cyclone IV Gx - EP4CGX150DF3117** FPGA as your target device. Part of your demo grade will be based on the maximum clock rate listed in the timing analyzer report. Frequencies above 45MHz will receive extra credit. Anything below 25MHz will receive a zero for the final demo.

Question 2.2.1: How many cycles does your program take to execute? (Note: it will be different than the last lab)

Question 2.2.2: At the end of the program, what are the contents of the following memory locations: 0x40000808, 0x4000080c, 0x40000810, and 0x40000814?

Question 2.2.3: At the end of the program, what are the contents of the following registers: $8, $9, and $31?

2.3: Hazard Table

Create a table that describes what hazards may occur when a certain instruction is followed by another instruction and what action you should take in order to deal with the hazard. Below are a few sample entries that should be include din your hazard table and should help you understand what your table should contain:

	R	**I**	**L**	**S**	**JR**	**LUI**	**B**	**JAL**
R	1a: exmem.rd=idex.rs(1) =idex.rt(2) 2a: memwb.rd=idex.rs(3) =idex.rt(4) 3a: memwb.rd=ifid.rs(5) =ifid.rt(6)				1a: idex.rd=ifid.rs(stall) 2a: exmem.rd=ifid.rs(8) 3a: memwb.rd=ifid.rs(5)			
I	1a: exmem.rt=idex.rs(1) =idex.rt(2) 2a: memwb.rt=idex.rs(3) =idex.rt(4) 3a: memwb.rt=ifid.rs(5) =ifid.rt(6)							
L		1a: ifid.rt=pre.rs(stall) 2a: memwb.rt=idex.rs(3) 3a: memwb.rt=ifid.rs(5)						
S	do nothing							
JR			do nothing					
LUI	1a: exmem.rt=idex.rs(11) =idex.rt(12) 2a: memwb.rt=idex.rs(13) =idex.rt(14) 3a: memwb.rt=ifid.rs(15) =ifid.rt(16)							
B					do nothing			
JAL		1a: exmem.31=idex.rs(21) 2a: memwb.31=idex.rs(23) 3a: memwb.31=ifid.rs(25)						

(1).	forward exmem.aludata to rs of alu
(2).	forward exmem.aludata to rt of alu
(3).	forward WBdata to rs of alu
(4).	forward WBdata to rt of alu
(5).	forward WBdata to rs of idex reg
(6).	forward WBdata to rt of idex reg
(8).	forward exmem.aludata to rs of idex reg
(11).	forward ui from exmem to rs of alu
(12).	forward ui from exmem to rt of alu
(13).	forward ui from memwb to rs of alu
(14).	forward ui from memwb to rt of alu
(15).	forward ui from memwb to rs of idex reg
(16).	forward ui from memwb to rt of idex reg
(21).	forward PC+4 from exmem to rs of alu
(23).	forward PC+4 from memwb to rs of alu
(25).	forward PC+4 from memwb to rs of idex reg

The first entry says that when an Rtype instruction is followed by another Rtype (both of which are not jump register instructions), the instructions are 1 instruction apart, and rd of the instruction from the

EXMEM register is equal to either the rs or rt of the instruction from the IDEX register, you should perform actions (1) or (2) to correct the hazard, which is forward the alu data from the EXMEM register to the rs or rt inputs of the alu accordiingly. The first entry also contains the information of what to do when the Rtype instructions are 2 or 3 instructions apart.

Section 3: Report
Each of the remaining assignments should follow this format:

1. The report should contain the following sections
   - Cover page – with your name, date, and assignment number
   - Introduction – briefly comment on the assignment and what it accomplishes
   - Lab3 Demo Program:
     - Annotated simulation of the output of your processor from the Lab3.mif file used in the final demo.
     - Answers to questions in the lab section.
   - Hazard table as described in this assignment
   - Appendix
2. The report needs to be typed. Annotations should not be scans of hand-drawn annotations and illegible annotations are almost as bad as no annotations.
3. Any waveform that is not labeled and annotated will not be considered as valid.
4. Make sure you use the names for inputs, outputs, and entities as outlined in the assignment. This will be essential for connecting these components together in later assignments.

Grade Breakdown

Section	**Deliverables**	**Percentage of Total grade**
Final Demo	Your processor running the Lab3.mif file demonstrated to be working	15%
2.1	Design of your multi-cycle processor with detailed diagrams	20%
2.2	Lab3.mif decode, similar to previous lab	45%
2.3	Hazard table as described in this assignment. Note that you will lose 1 point per conceptual error in the table.	25%
Report	Report follows the format given in section 3 (1%). Report is neat and professional (1%).	5%